REMARKS

Status of the Claims

- Claims 1-6 and 8-23 are pending in the Application after entry of this amendment.
- Claims 1-23 are rejected by Examiner.
- Claims 1, 11, 17, and 21 are amended.
- Claim 7 is cancelled without prejudice or disclaimer.

Claim Rejections Pursuant to 35 U.S.C. §102

Examiner has rejected Claims 1-23 under 35 U.S.C. §102(e) as being anticipated by U.S. Pat. No. 6,934,712 to Kiernan et al. Applicant respectfully traverse the §102(e) rejection.

Kiernan et al. teaches:

> A method for publishing relational data as XML by translating XML queries into queries against a relational database. Conversion of the relational database into an XML database is not required. Each relational table is mapped to a virtual XML document, and XML queries are issued over these virtual documents. An XML query is parsed and transformed into a language-neutral intermediate representation, which is a sequence of operations describing how the output document is derived from the underlying relational tables. The intermediate representation is then translated into an SQL query over the underlying relational tables. The intermediate representation is also used to generate a tagger graph, which the tagger runtime 'walks' to generate the tagged, structured XML output. Each of the nodes of the tagger graph are operators which perform processing on the results of the SQL query. The SQL query is executed, and the SQL query results are then provided to the tagger. The tagger runtime applies the operators of each node to the inputs at that node to produce the structured XML document as a query result, guided by the structure of the tagger graph. (Abstract).

In the Final Office Action dated 7/12/2006, page 10, item 7 part b, the Examiner states:

> "b. Applicants' argument stated as "the tagger graph of Kiernan et al. cannot be the semantic (intermediate) representation because Kiernan et al. requires that the intermediate representation to exist before a tagger graph is generated" In response to the Applicants' argument, the tagger graph is not the intermediate representation but rather what is generated from the intermediate representation. The operators from the query are represented in the intermediate representation and are generated by the parser. (Figure 2 reference 210)."

Applicant agrees with the Examiner. Applicant has amended Claim 1 to recite:

A method for semantic representation of one or more XML language inquiries across relational and non-relational data sources comprising:
receiving at least one inquiry;
defining a plurality of nodes of a graph structure which represents the at least one inquiry, the graph structure having at least one node object for every operation within the at least one received inquiry;
translating each of the at least one node objects using operators; and
generating a semantic representation having the graph structure;
wherein the semantic representation explicitly describes a meaning of the one or more XML language inquiries and wherein the semantic representation decouples front-end language compilers from back-end query engines that use the semantic representation.

Whereas Claim 1 recites that the semantic representation has a graph structure, Kiernan's intermediate representation is not a graph structure. Indeed, as affirmed by the Examiner (see above), the tagger graph is not part of the intermediate representation but is generated only after the Kiernan intermediate representation is complete. This is also evident by viewing Figure 2 of Kiernan which shows the intermediate representation present at the output of the parser 210 and the rewrite engine 220. The tagger 240 is only useful after the intermediate representation if translated to a SQL query by the translator 230, the SQL query is executed, and results returned from the database management system 102 to the tagger 240.

Also, as taught in Kiernan at col. 7 lines 59-64:

"The tagger graph is generated from the intermediate representation and, therefore, reflects the structure of the resultant XML document. The tagger runtime 240 walks the tagger graph applying the operators of each node to the inputs at that node to produce the structured XML document as a query result, guided by the structure of the tagger graph." (col. 7, lines 59-63).

Thus, the Kiernan tagger graph "reflects the structure of the resultant XML document" whereas the semantic representation of Claim 1 describes a meaning of the one or more XML language inquiries. Applicant notes that "the one or more XML language inquiries" of Claim 1 are inputs whereas the "resultant XML document" in Kiernan are outputs. The Kiernan intermediate representation does not have a graph structure as recited in Claim 1. Applicant submits that Kiernan does not disclose any graph structure other than the tagger graph structure. Consequently, Kiernan is without a teaching that the intermediate

representation has a graph structure. Accordingly, Kiernan does not disclose all of the elements of amended Claim 1.

There are also other differences between Kiernan and amended Claim 1. Figure 2 of Kiernan teaches that a schema mapper 200 links between the parser 210, which produces the intermediate representation of the Relational Database Management System 102 to perform the search. According to Figure 2, the schema mapper 200 inputs relational schema from the relational database 102 and produces a default view as an input to the parser 210. The parser 210 thus inputs the default view of the relational database and the input XML query to produce an intermediate representation.

This is highlighted in the Kiernan-provided definition of the intermediate representation. The definition provided by Kiernan is:

> "The Intermediate Representation
>
> The intermediate representation is a sequence of parameterized operations that describe how the output of the XML-QL query is derived from the underlying relational tables." (col. 5 lines 56-60).

Applicant respectfully submits that Kiernan has correctly defined his intermediate representation because the definition includes the aspect of the "underlying relational tables" of the relational database 102. Kiernan includes aspects of the underlying relational tables in the intermediate representation because of the connection that the schema mapper 200 provides between the parser input 210 and the relational database engine 102.

Applicant notes that amended Claim 1 not only recites that the semantic representation has a graph structure which is absent from Kiernan, as discussed above, but also, Claim 1 produces a semantic representation that describes the meaning of the input inquiry. This contrasts with Kiernan because the intermediate representation of Kiernan also includes a description concerning "how the output of the XML-QL query is derived from the underlying relational tables". Amended Claim 1 recites no such connection with any underlying relational tables as Kiernan does.

One aspect of the current invention is that the semantic representation of the input queries decouples the front-end compilers from the back-end query engines that use the semantic representation. (See paragraph 0047 of the originally filed application). Applicant has amended Claim 1 to include this feature. Referring to Figure 2 of Kiernan, Applicant

submits that Kiernan teaches linking the front-end parser 210 with the back-end SQL query engine in the relational database management system 102. Kiernan performs this linkage of front-end compiler to back-end query engine by providing data from the RDBMS 102 via the schema mapper 200 to the front-end parser 210 which creates the intermediate representation. Thus, the front-end and the back end of the Kiernan system are linked and that linkage allows the intermediate representation to be "a sequence of parameterized operations that describe how the output of the XML-QL query is derived from the underlying relational tables" as stated in Kiernan col. 5 lines 56-60. Amended Claim 1 decouples front-end language compilers from back-end query engines that use the semantic representation. Thus, Kiernan actually teaches away from amended Claim 1.

In review, whereas Kiernan teaches that the intermediate representation is without a graphical structure, amended Claim 1 recites that the semantic representation includes a graphical structure. Therefore Kiernan fails to teach a semantic representation having a graph structure. Kiernan fails to teach all of the elements of amended Claim 1.

Whereas Kiernan teaches linking a back-end query engine (inside the RDBMS 102 in Figure 2) to the front-end compiler (parser 210 of Figure 2), amended Claim 1 recites that "the semantic representation decouples front-end language compilers from back-end query engines that use the semantic representation". Thus, Kiernan actually teaches away from amended Claim 1 and is thus not an effective reference to teach all of the elements of amended Claim 1.

Applicant has amended independent Claims 11, 17, and 21 to include the aspects of a semantic representation that contains a graphical structure and that the semantic representation decouples front-end language compilers from back-end query engines that use the semantic representation.

Since Kiernan fails to teach or even suggest that a semantic representation includes a graphical structure and since Kiernan teaches away from the semantic representation decoupling front-end language compilers from back-end query engines that use the semantic representation, then Kiernan cannot anticipate amended independent Claims 1, 11, 17, and 21 and their respective dependent claims.

Applicant therefore respectfully requests withdrawal of the 35 USC §102(e) rejection and submits that Claims 1-6, and 8-23 patentably define over the cited art because all elements of the amended independent Claims 1, 11, 17 and 21 are not found in the cited art.

Claim Rejections Pursuant to 35 U.S.C. §103

Claim 9 stands rejected under 35 U.S.C. §103(a) as unpatentable over U.S. Pat. No. 6,934,712 to Kiernan et al. in view of U.S. Patent No. 6901,410 to Marron et al. Applicant respectfully traverses the rejection.

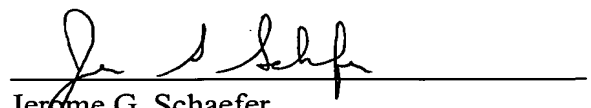
Claim 9 ultimately dependent on independent Claim 1. As mentioned above, Kiernan et al. does not disclose all of the elements of independent Claim 1. Accordingly, a prima facie case of obviousness is not established with respect to Claim 1 and its dependent claims. Applicant respectfully requests withdrawal of the 35 U.S.C. §103(a) rejection of Claim 9 as it patentably defines over the cited art.

Conclusion

Applicant respectfully requests reconsideration of all pending claims in light of the amendment and discussion above. Applicant respectfully submits that all pending claims patentably define over the cited art.

Respectfully Submitted,

Date: August 23, 2006



Jerome G. Schaefer
Registration No. 50,800

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439